Brian D. Ballentine
brian.ballentine@mail.wvu.edu

Composition in the Freeware Age: Assessing the Impact and Value of the Web 2.0 Movement for the Teaching of Writing

Hacker Ethics & Firefox Extensions: Writing & Teaching the 'Grey' Areas of Web 2.0

This article adds "hackers" and "hacker ethics" to the growing complexities of the ongoing debate over Web 2.0 as both a term and concept. Contrary to the pejorative connotation the word often carries, free software pioneer Richard Stallman defines a hacker as "'[s]omeone who loves to program and enjoys being clever about it'" (p. 53). The reach and meaning of the term, however, have spread beyond the world of computing. Programmer and open source activist Eric Raymond's online Jargon File contains multiple entries for "hacker" including: "An expert or enthusiast of any kind. One might be an astronomy hacker, for example." According to Raymond a hacker is also: "One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations." The first objective of this text is to create a link between the ballooning definitions of writing to the activities and motivations of a hacker. Writing has moved deeper into what Christopher Thaiss (2001) has called the "multimedia swamp" where creative expressions such as video remixing and other forms of digital authoring are not only classified as writing but being taught in a variety of composition classrooms experimenting with new media. Therefore, I wish to posit that as writers in Web 2.0 environments we are all in a very real sense hackers. While I have initiated the claim elsewhere that it is becoming more difficult and less fruitful to distinguish between programmers, designers, and authors and their melding roles in digital spaces (Ballentine), this text and ensuing argument focuses on using specific Web 2.0 technologies as a backdrop for discussion on what we as writers/hackers can glean from "hacker ethics." Specifically, I will be pursuing what we can learn from the hacker community about how to teach students to write ethical Web 2.0 texts.

While the term "ethics" has received some scholarly attention in networked communities and is prevalent among users and developers of free and open source software, discussions

regarding the term are typically understoond in the context of an individual's duty to make their work available to the rest of the community (Himanen, 2001; Raymond, 2000; Stallman, 1999; Weber, 2004). The free and open source mantra of "information wants to be free" is shared by Web 2.0 not just in the fact that many popular Web 2.0 applications are open source but that popular Web 2.0 publication and distribution methods such as RSS feeds, social book-marking, and application programming interfaces (APIs) share the same philosophy for sharing and manipulating content. This is not to say that there are no Web 2.0 projects developed for commercial gain. Instead, as we will see in the discussion in the section on Web 2.0 and Open Content, Web 2.0 succeeds because, "Giants like Yahoo and Google have thus far taken a mostly nonproprietary stance toward their data" (McHugh, 2008, p. 139). And without suggesting that Web 2.0 technologies are inherently benevolent, it is the unregulated treatment of data that fuels both commercial and noncommercial innovation in this arena. Looking once again to Raymond's Jargon File, we find a parallel between Web 2.0 and his first entry for "hacker ethic" which states: "The belief that information-sharing is a powerful positive good, and that it is an ethical duty of hackers to share their expertise by writing open-source code and facilitating access to information and to computing resources wherever possible."

Ethical concerns as they relate to writing and technology are of course not new. Composition, multimedia writing, and professional communication classrooms, where the influence of digital advancements such as Web 2.0 continue to be felt, offer environments for "keeping ethics, rhetoric, and writing together as an intertwined set" (Porter, 1998, p. 23). Ultimately I wish to demonstrate that pedagogically we do not have a choice. As James Porter (1998) remarked, "When one writes, one decides. Writing is an action involving an ethical choice about what one is to be and what one is to do. At the point when you begin to write, you begin to define yourself ethically" (p. 150). Porter's book, *Ethics and Internetworked Writing*, which is now more than 10 years old, contained the warning that "we are in an ethically sensitive and important time right now because what we as users (and as teachers of users) do on the networks

will help constitute the norms for such discourse as they become stabilized and legally sanctioned (or not) in the future" (p. 8). Web 2.0 has only ratcheted-up the sensitivity levels by enabling a larger group of users to engage with more data and to test the limits of how and how much of that data is used. Because an increasing number of us use Web 2.0 technologies and in essence have joined a community of writers/hackers, this text advocates moving beyond just modeling ethical online behavior and using the classroom as a space to introduce hacker ethics. To do so, this text focuses on two new Web 2.0 extensions (Web Developer and Greasemonkey) for the open source browser Firefox that I believe extend an unprecedented invitation to remix, hack, manipulate, and even sabotage content. These two extensions and the activities they enable are discussed within the framework of white, black, and grey "hat" hacker ethics.

First, this text attempts to outline how and why we are essentially hackers when we write and create in digital spaces. Next, since new modes of composition require special consideration in relation to Web 2.0, I provide a discussion on primary characteristics of Web 2.0 as well as its relationship to open source. The subsequent section will then attempt to parse the complexities of hacker ethics and demonstrate how the field of composition's growing concerns regarding academic integrity and plagiarism mirror some of the challenges in attempting to define what constitutes ethical hacking. I propose that writing, or the writing/hacking that we and our students do with Web 2.0 technologies, can benefit from considering the ethical hacking color spectrum of white, black, and gray "hat" hacking and their associated behaviors. With this foundation in place, I provide an overview of Firefox extensions before introducing two specific Web 2.0 technologies, Web Developer and Greasemonkey, and their potentials for pairing writing and hacker ethics in the classroom.

Writing as Hacking

As Web 2.0 takes us deeper into Thaiss' (2001) "multimedia swamp," he also wonders "where does the 'writing' end and something else take over?" (p. 306) Indeed, our profession

struggles to keep pace with the latest production technologies and the new generic artifacts that may or may not fall under our purview. In, “Made Not Only in Words: Composition in a New Key,” Kathleen Blake Yancey (2004) states that, “Never before have the technologies of writing contributed so quickly to the creation of new genres” (p. 298). This leads her to ask, “What is writing, really?”

> It includes print: that seems obvious. But: Does it include writing for the screen? How visual is it? Is it the ability to move textual resources among spaces…? Is it composing…not only about medium but also specifically about technology? Suppose I said that basically writing is interfacing? (p. 299)

Intended or not, the term “interface” is a loaded one in the context of writing in digital spaces. It is at once the act of communicating as well as a framed area facilitating and defining where and how the communication transpires. Not long ago the options for those designing and using digital interfaces were few. Web 2.0 has broadened interfacing for both the designer and the user, effectively blurring those once separate roles. As a result, user expectations of quality, usability, and overall effectiveness and persuasiveness of the communication space have increased to the point that “the cost of building a mediocre interface is higher than it used to be” as “frustrated users can give up” and easily go elsewhere (Tidwell, 2006, p. xi). As Stuart Selber (2004) remarks in his *Mulitliteracies for a Digital Age*, interface design is a “rhetorical endeavor” (p. 28). He posits that “interface design problems are more like writing than programming problems and that although all projects have technical aspects, mathematical and scientific formalisms are inadequate in design situations that involve social concerns and interactions” (p. 28). As an added caveat, I will demonstrate how Web Developer and Greasemonkey empower the frustrated user to usurp the designer role and rewrite or hack the inadequate interface. In addition to the social concerns embedded in hacking an interface, we will not be able to overlook the responsibility of evaluating the ethics of such actions.

This phenomenon of re-writing or hacking the interface supports what noted legal scholar and copyright reform activist Lawrence Lessig refers to as “read-write culture.” Over the last

several years Lessig has given numerous lectures on intellectual property law, the internet, and their relationships to our (in)ability to legally "remix" digital content. Rhetoric and composition instructors may have seen Lessig's presentation "Remix Culture" at the 2005 Conference on College Composition and Communication or perhaps viewed his March 2007 Technology Entertainment Design (TED) online lecture titled "How Creativity Is Being Strangled by the Law." While the law as it regulates the world of print permits read-write culture by allowing for the practice of "remix," Lessig and others worry that copyright term extensions and other adjustments to intellectual property law are pushing us in the direction of read-only culture. In his TED lecture, Lessig notes that read-write culture and our legal ability to remix digital content means "people participate in the creation and the recreation of their culture." The consumer moves from a passive role into the position of creator or a co-creator of content. Lessig is clear, however, that he is not advocating for the mere copying and redistribution of other people's copyrighted materials.

> I'm talking about people taking and recreating using other people's content using digital technologies to say things differently...now the importance here is not the technique…the importance is that the technique has been democratized. It is now anybody with access to a $1500 computer who can take sounds and images from the culture around us and use it to say things differently. These tools of creativity have become tools of speech. It is a literacy for this generation. This is how our kids speak. It is how our kids think. It is what your kids are. (2007)

A number of composition scholars have initiated thoughtful conversations on technology and literacy (Gurak, 2001; Selber, 2004; Selfe, 1999) and we can benefit from that research as we begin to approach remixing and hacking as a form of writing and literacy. Cynthia Selfe defines "technological literacy" as "a complex set of socially and culturally situated values, practices, and skills involved in operating linguistically within the context of electronic environments, including reading, writing, and communicating" (p. 11). Similarly, Information and Communication Technologies or ICT has been an educational buzzword for several years and beyond just

installing networked computers in a classroom, ICT research suggests that a "crucial" part of the "changing role of the teacher" will be to use available technologies to "encourage critical thinking skills, promote information literacy, and nurture collaborative writing practices" (Wheeler, 2001, p. 13). Critical thinking skills are a key component to Selber's three-part "mutliliteracy" comprised of what he terms functional, critical, and rhetorical literacy. According to Selber, incorporating critical literacy in the classroom means that we must "invite students to approach an artifact with inquiries about it that are different from the ones directly imagined by author-to-readers intention structures, making available an oppositional discourse that can be used to critique a dominant discourse" (p. 97). At its best, critical literacy translates into students using digital technologies to react to dominant discourses with "technological adjustment and technological reconstitution" (p. 104). Among the outcomes of technological adjustment, users have the opportunity to "engage in micropolitical acts of modification that adapt technologies to users" (p. 105). Technological reconstitution is an even more "aggressive response" where users may "create counter artifacts that displace the politics of technological regularization" (p. 105).

For example, many web sites track how their visitors get to their sites by way of an "HTTP referer" [sic]. Depending on what link a user clicks on to get to a web site (that is, how they are referred to their content) a user may gain or be denied access to different materials. For over a year now, the *Wall Street Journal* has allowed its online readers to use the Web 2.0 link aggregator site digg.com to link or "digg" its articles. Any article that has been "dugg" may be clicked on and read for free so long as Digg is the referrer. This includes articles that would normally be restricted to paid subscribers. While the *Wall Street Journal* was using Digg's social networking power to boost attention to its articles, hackers eager to exploit an opportunity to work around the $79.00 online subscription fee created a browser extension called "refspoof" in order to make it appear that any *Wall Street Journal* article has been referred by Digg. Those in favor of this technological reconstruction are quick to point out that "it's completely legal" but also note that it is "slightly deceptive" (Manjoo 2008). What we will witness in the sections

introducing Web Developer and Greasemonkey will be similar instances of hacking/writing that are realizations of theoretical concepts like Selber's technological adjustment and reconstitution.

Web 2.0 & Open Content

In many ways, Web 2.0 innovations like "refspoof" are embodiments of the theories Selber describes in his chapter on critical literacy. Because content owners have chosen to take a more hands-off approach to controlling access to their data, hackers/writers are seizing opportunities to re-write content to produce any number of counter artifacts. Web 2.0 is the realization that we are no longer passive roamers of the web, refining our search parameters and clicking links with a hope its path yields our desired result. Web 2.0's contribution to read-write culture has been, in part, to set a precedent for utilizing unfettered content found online on a much larger scale. The right to view and use this ever growing mass of information has been the fuel for Web 2.0. The practice of keeping content open is critical to read-write culture because as Michael Salvo (2005) points out, "Access to the database determines who gets to speak, as well as who has the authority of the data behind the words. Design of the database determines who gets access" (p. 65). Fortunately, the model design for Web 2.0 databases has been geared predominately toward openness.

Tim O'Reilly's "What is Web 2.0?" (2005) is perhaps the most widely recognized recounting of Web 2.0's emergence. Shortly after the dot-com bubble burst in 2001, O'Reilly became aware that ethical views on sharing data as well as the practice of involving users in development processes encouraged by early hackers like Stallman and Raymond were being embraced to a much greater degree. The landscape of the internet was changing. Instead of designing applications that blocked users from valuable data, Web 2.0 technologies promoted read-write culture by soliciting user participation. Consequently, Web 2.0 can "harness collective intelligence" from all over the internet and view users as co-developers (O'Reilly, 2005, p. 2).

Again, marshalling the brain-power of many to attack a common problem is not new to hacker communities.

Back in August of 1991, Linus Torvalds, the developer of the popular open source operating system Linux, posted the first version of his project online as an invitation to what would become thousands of co-developers (Goetz, 2003, p. 164). Since that time, Torvalds has made a habit of distributing new versions of Linux's code with an unprecedented frequency and transparency. Raymond became involved with Linux in 1993 and by 1997 he debuted the first version of his manifesto "The Cathedral and the Bazaar" at a Linux conference. In very practical terms, Raymond writes: "Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging" (The importance of having users). Prior to observing Torvalds manage the development of Linux, Raymond "believed that the most important software needed to be built like cathedrals, carefully crafted by individual wizards or small bands of mages working in splendid isolation, with no beta released before its time" (The cathedral and the bazaar). In contrast, Linux blossomed due to a flurry of innovative ideas contributed to an ever growing and collective knowledge base all centered on advancing Linux. Raymond remarks:

> Linus Torvalds' style of development – release early and release often, delegate everything you can, be open to the point of promiscuity – came as a surprise. No quiet, reverent cathedral building here – rather, the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches out of which a coherent and stable system could seemingly emerge only by a succession of miracles. (The cathedral and the bazaar)

Similar to the unfettered content of Web 2.0, the source code for Linux is still found circulating freely for anyone to download, use, or contribute to as a co-developer. Normal practice on the part of the cathedral builders would entail locking down the source code with intellectual property law but Linux uses the law to instead promote openness and access to code. This approach to content and the subsequent growth of a project like Linux coincide with what

O'Reilly describes as one of the primary tenets of Web 2.0 where any service offered by a Web 2.0 site "automatically gets better the more people use it" (p. 2). In short, collaboration begets success.

Icons of the Web 2.0 phenomenon like Google have made their fortunes on collecting metadata from millions of web sites and serving it up in neatly refined search results. According to O'Reilly, Google is a success because it recognizes that the business models, particularly approaches to content and collaboration, have evolved differently online. "Google isn't just a collection of software tools, it's a specialized database. Without the data, the tools are useless; without the software, the data is unmanageable" (p. 1). Google now offers a suit of Web 2.0 services to make use of that data including their AdSense program. When a web site owner enrolls in the AdSense program, Google "serves" or includes advertisements on the web site owner's pages which are dynamically generated based on a number of variables including the page's content and the visitor's geographic location. Google pays participants based on the number of clicks generated by each advertisement. A recent article on nichegeek.com profiles several individuals making hundreds and even thousands of dollars a month with this Web 2.0 service. By comparison, a Web 1.0 model would see Google advertising only on its own search pages. AdSense means Google's rich database puts more information in front of more users including even those that did not begin their browsing session at Google.com. In short, Google gives up some of their control and shares profit in order to increase their advertising reach.

In addition to Google, sites like Flickr, Flock, BitTorrent, and services by Yahoo! invite users to remix, hack, or otherwise repurpose "their" collected data. Yahoo! Pipes, for example, is billed as "a powerful composition tool to aggregate, manipulate, and mashup content from around the web." After a user creates or "composes" a Pipe, the software allows users to save it and even share it with others. My last visit to the Yahoo! Pipes page featured a Pipe for remixing Yahoo! Search results for Napa Valley wineries with photos of the same Napa wineries posted to Flickr.

This Pipe, like other Web 2.0 remixes, becomes more robust the more people use and contribute their images to Flickr.

How are these hacks possible? At its core, Web 2.0 relies on the concept that the web itself is our universal computing platform. That is, we are not writing and developing for Windows, Leopard, or even one of the various Linux distributions but instead a platform with open standards and protocols free from any one agenda. The idea is to leverage the web as platform to reach more users without the typical practice of attempting to control that platform. This absence of control means the end result is that we as users or co-developers can follow O'Reilly's advice to "design for 'hackability' and remixability" so that the "barriers to re-use are extremely low" (p. 4).

An excellent explanation of leveraging the web as a platform is found in the often cited YouTube sensation piece, "The Machine is Us/ing Us" by anthropology professor Michael Wesch. With Web 1.0, developers used HTML to "mark up" content as a means to format and display data. Using HTML resulted in fixed content that was bound up with its form. As Wesch explains, a newer method for tagging content using extensible mark-up language or XML gives users the flexibility to describe the content without prescribing its form. Separating form from content means that the content can, for example, be exported via RSS feeds and aggregated as a user wishes. Wesch also reminds us that users do not need to know "complicated code" in order to participate and use Web 2.0 technologies. With Flickr, users can post and tag images (e.g., describing an image of a Napa Valley winery) and that same image may ultimately be remixed and shared in a variety of uses including search results on Yahoo! In answering his own prompt of "Who will organize all of this information?" Wesch types in response that we all will. However, the close of his video offers a series of serious challenges to us, the users of Web 2.0. Wesch writes that "we will need to rethink a few things" and his list of considerations includes "copyright," "authorship," and "ethics."

Hacker Ethics

> The word 'hacking' is sexy, exciting, seemingly seedy, and usually brings about thoughts of complex technical activities, sophisticated crimes, and a look into the face of electronic danger itself. (Harris et al., 2005, p. 11)

There are many among hacker communities that bemoan the vilification of a word that was intended, as Raymond claims, to signify someone who thrives on "intellectual challenge." Even if we agree that the most suitable definition for a hacker is "One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations" the debate over what behaviors are ethically sound when getting around those limitations remains. The discussions from the previous sections of this text have revealed some of the ethical expectations regarding behavior and conduct in open source communities and Web 2.0 where content is shared and collaboration valued. Once again the open source community can assist in this attempt to pin down a multi-faceted term such as hacker ethics. Returning to Raymond's online Jargon File we find two entries on "hacker ethic" (the first was introduced earlier) which serve as good points of entry to unpacking what the open source community understands as ethical behavior and conduct. Raymond writes:

1. The belief that information-sharing is a powerful positive good, and that it is an ethical duty of hackers to share their expertise by writing open-source code and facilitating access to information and to computing resources wherever possible.
2. The belief that system-cracking for fun and exploration is ethically OK as long as the cracker commits no theft, vandalism, or breach of confidentiality. (Hacker ethic)

Notice that Raymond uses the term "cracker" instead of "hacker" for someone who breaks into a system. Cracker is "a term created by hackers in the mid-1980s to differentiate the two communities" (Attackers and vulnerabilities). The term has not achieved wide-spread use especially outside of the more specialized IT or programming communities. Regardless of what

we label an activity we are left ultimately with the same debate over what falls into the category of ethical hacking versus malicious cracking.

Raymond's first entry aligns with the spirit and ideology of the open source movement and Web 2.0. For those using Web 2.0 applications and all of their open data, the position that we have an ethical responsibility to continue to promote that openness seems less than controversial. It is not hard to imagine, however, IT professionals and system administrators employed to keep corporate networks and content secure, taking umbrage to the idea of hackers compromising their systems for amusement just because they are armed with the notion that such practices are ethically "OK." As we will see in the coming discussions on the texts *Gray Hat Hacking* (Harris, et al., 2005) and *Certified Ethical Hacker* (Gregg, 2006), Raymond's second entry does not have universal agreement. However, his own leanings are revealed in his remark that "the belief that 'ethical' cracking excludes destruction at least moderates the behavior of people who see themselves as 'benign' crackers." While excluding malicious activities from the definition of ethical hacking may appease members of the open source or Web 2.0 communities, the practice of "benign" hacking does not fall into the realm of acceptable actions for most IT professionals.

In an effort to dispel any "sexy" intrigue attached to hacking, Harris (2005) and the other authors of *Gray Hat Hacking* remark that in many ways, the malicious hacking of breaking into networks or stealing content is just a different breed of theft and sabotage only with a new array of tools. Indeed, what is interesting about books like *Gray Hat Hacking* and others such as Michael Gregg's (2006) *Certified Ethical Hacker* that purport to have the average IT professional as their intended audience, is that they are very clear that ethical hacking means getting owner authorization before hacking anything. According to Gregg, "Ethical hacking is a form of legal hacking that is done with the permission of an organization to help increase its security" (p. 20). Whether as a company employee or an outside consultant, all hacking requires permissions and full disclosure. While there is discussion dedicated to proper disclosure and what a hacker needs to know about our legal system, these types of texts (Gregg's text actually helps IT professionals

prepare for the International Council of Electronic Commerce Consultants ethical hacking certification exam) confront quickly issues of behavioral ethics and then move on to technical processes and procedures for hacking. That is, if you are interested in how to run penetration tests and basic Windows and Linux exploits these texts are the appropriate resources. If you want a philosophical debate over information sharing these texts are less useful. Perhaps because Gregg allots for no ambiguity or "gray" areas regarding ethical behavior for hacking (either you have permission or you do not), discussions on conduct are brief. Gregg has a section dedicated to what he calls "Rules of Engagement" where the third bullet point reads: "Be ethical."

> Be ethical – That's right; the big difference between a hacker and an ethical hacker is the word ethics. Ethics is a set of moral principles about what is correct or the right thing to do. Ethical standards are sometimes different from legal standards in that the laws define what we must do, whereas ethics define what we should do. (p. 33)

It is also worth noting that both Gregg's book and *Gray Hat Hacking* are exceptionally sensitive to the skills they are teaching. Gregg notes that, "Nothing contained in this book is intended to teach or encourage the use of security tools or methodologies for illegal or unethical purposes. Always act in a responsible manner" (p. 20). Harris, et al., defend their book by stating, "the ethical hacker has to know what the bad guys are using, know the new exploits that are out in the underground, and continually keep her skills and knowledge base up to date" (p. 12). The authors cite the "educational piece" provided by books such as theirs that makes "the difference between hacking and ethical hacking" (p. 12). What is surprising is not so much that these books exist in the first place but that they differ so drastically from the philosophical and ideological designs for sharing knowledge found in the writings of Raymond and even O'Reilly.

Granted, most of us teaching writing are not hacking into someone else's computer to install spy-ware or orchestrating a denial of service attack on an ill prepared server. However, texts that purport to teach "proper" ethical hacking do so by creating a binary of "good" versus "bad" behaviors similar to that of our plagiarism policies often used to teach or enforce our own

academic writing/hacking. As writing instructors we often use texts that approach issues of plagiarism with the same binary. In *Writing That Works*, (Oliu, Brusaw, Alred, 2007) a popular textbook for professional and business writing courses, plagiarism is cast, as it typically is, as stealing. "Plagiarism is considered to be the theft of someone else's creative and intellectual property and is unacceptable in any field. If you intend to publish, reproduce, or distribute material that includes quotation from published works, including Web sites, you may need to obtain written permission from the copyright holders of those works" (Oliu, Brusaw, Alred, p. 178). While copyright violations and plagiarism are different punitively, definitions such as these conflate the act of holding up someone else's work as your own with stealing intellectual property. The larger problem is, of course, that if we adhere to the letter of existing copyright law then permission is almost always required. This creates what Lessig describes as permissions driven culture where the law becomes an impediment or a barrier to creativity and a "burden to innovation" (2004, p.193). Given the type of writing students do with Web 2.0 technologies, Lessig's question for composition instructors during his "Remix Culture" presentation, asking whether or not writing will continue to be "allowed," appears more relevant than ever.

Hacking, Writing, & Plagiarism

In "Framing Plagiarism," Linda Adler-Kassner, Chris Anson, and Rebecca Moore Howard note that what is allowed is complicated because "in no community is the textual value system unitary or stable" (2008, p. 240). The scholars are critical of news stories and articles that simplify current issues of textuality and plagiarism by casting student writers as mere cheats. The unfortunate result is a shift in emphasis from writing instruction, including discussions on value systems relating to "text," to how to catch cheaters. In her 1999 work, *Standing in the Shadow of Giants: Plagiarists, Authors, Collaborators,* Rebecca Moore Howard challenges the "criminalization" of plagiarism and asks that we reevaluate what practices the term encompasses. Much like Lessig's views on the ills of thoughtless copying and redistribution of copyrighted

materials, Howard does not condone students taking content outright and passing it off as their own. However, as students evolve as writers, they attempt to situate themselves within their respective discourse communities in part by appropriating and incorporating other sources of material. Good writing instruction focuses not on catching cheaters but on learning and respecting the value systems of different communities. Similar to a Web 2.0 remix, Howard appreciates what she terms "patchwriting" outcomes as unique and valid artifacts. Just as Lessig suggests with remix, as a patchwriter, students are taking materials from culture around them and using that existing content "to say things differently."

In a more recent article, "Understanding 'Internet Plagiarism,'" (2007) Howard draws on postmodern theory to set the groundwork for an argument that "all writing is relational and intertextual" (p. 9). As such, we are situated amidst a textual revolution with which we can either critically engage and embrace or pull away from with efforts to lock down and control access to texts. According to Howard, the latter is a waste of our time. "It is no longer possible to control access to text, and it is no longer possible to imagine that writers do not draw copiously on other texts, both consciously and unconsciously" (p. 10-11). Appropriation and intertextuality are inherent in writing. For evidence, Howard cites Geisler et al. and their scholarship on "IText" or "information technologies with texts at their core" (p. 270). The authors cite everything from email to the original ARPANET to the Palm Pilot as IText examples. While "IText" predates Web 2.0, their description of it is remarkably prescient. They write: "People use IText documents as part of larger activities, carrying out meanings with motives and using tools that are built on prior activities and activity systems but transforming them in their new electronic contexts" (p. 273). This transforming or what can now be called the hacking or remixing of texts and the resulting artifacts are embodiments of intertexuality that should not be dismissed as mere plagiarism. In order to combat what Howard calls the "widespread hysteria" brought on by the perceived threat of internet plagiarism, students would be best served by eliminating "inclusive and simplistic" labels that encompass otherwise productive patchwriting (p. 12, 13). For

situations where a student's work is questionable ethically, Howard suggest we pause before branding someone a plagiarist and instead consider this advice from the Council of Writing Program Administrators and its Statement on Best Practices:

> Ethical writers make every effort to acknowledge sources fully and appropriately in accordance with the contexts and genres of their writing. A student who attempts (even if clumsily) to identify and credit his or her source, but who misuses a specific citation format or incorrectly uses quota-tion marks or other forms of identifying material taken from other sources, has not plagiarized. Instead, such a student should be considered to have failed to cite and document sources appropriately. (Defining and Avoiding Plagiarism)

Of course, we are still exploring how we write in digital environments and that means constructing not only new citation methods but new pedagogy for the classroom. Since Web 2.0 at its best is, as O'Reilly suggests, designed for hackability and remixability, users are producing new digital text that were never before imagined and any formal documentation methods are naturally absent.

In a bit of irony, for all of the anxiety over internet plagiarism the composition community has grown fond of open source and its ethical approaches to knowledge sharing. Specifically, the National Council of Teachers of English and the Conference on College Composition and Communication passed recently a resolution on the adoption and use of open source software (2008) which reads, "BE IT THEREFORE RESOLVED that the Conference on College Composition and Communication support consideration of and strategic use of open source software whenever possible; will explore the use of open source software within its own organization and recommend that educators, institutions, and other educational organizations do the same." Of the reasons listed for pursuing open source, many are practical including claims that open source may spare technology budgets and prevent universities from being tied to a specific software vendor with no way out of a long term contract. Relevant to this discussion, however, is the additional claim that: "The open source development model parallels the

academic model of knowledge creation and distribution." While the resolution is a brief document and does not elaborate on how exactly academic and open source communities are alike, I read the claim as a direct acknowledgement and even endorsement of the intertextuality and ITexts Howard, Geisler, and others describe. The unanswered question remains: "How do we teach students to create ITexts that are ethical?" (Geisler, et al., p. 283)

Here I am suggesting that we take a cue from factions of the open source community that acknowledge the ethical gray areas of writing/hacking. Specifically, the online security guide documentation offered by Red Hat, a commercial distribution of Linux that profits by selling professional support and training for an otherwise free operating system, contains a loose but useful framework for hacker ethics. Red Hat does not pretend that achieving a universal definition of ethical hacking is possible and instead, the practice of hacking in any form is discussed within the spectrum of white, black, and gray "hat" hacking. The "hat" a hacker wears and its associative color is simply a metaphorical reference to the type of activity he or she is engaged with at a particular moment in time. Not surprisingly white symbolizes good or pure activity, black denotes bad or evil, and gray stands in for an ambiguous blend of the two.

According to Red Hat, "white hat hackers crack their own systems or the systems of a client who has specifically employed them for the purposes of security auditing" (Attackers and vulnerabilities). This permission centered definition is in accordance with ethical hacking as described earlier by Gregg and Harris, et al. For a student composing a Web 2.0 text by any name, that is, a remix or IText, white hat hacking would be analogous to that student systematically documenting all of his or her sources and acquiring all of the necessary permissions before using any digital content found online. Content for which permission was not granted (including cases when the "owner" of that content could not be located) would need to be omitted from the composition in order to comply with this definition of white hat writing/hacking.

Conversely, those that hack for ignoble purposes such as sabotaging servers or stealing confidential content are labeled “black hat hackers.” Black hats “are less focused on programming and the academic side of breaking into systems” (Attackers and vulnerabilities). Here, ethical writing/hacking would amount to a student acquiring content online and attempting to take credit for that content “as is” and as if it were his or her own work. Black hat writing/hacking means the student has no critical engagement with the content. That is, there is no “technological adjustment” or “technological reconstitution” of the artifact that Selber describes as important components of a student’s critical literacy education.

In practice, hacking is often more “grey” than the white or black binary. According to Red Hat’s security guide, “The grey hat hacker, on the other hand, has the skills and intent of a white hat hacker in most situations but uses his knowledge for less than noble purposes on occasion. A grey hat hacker can be thought of as a white hat hacker who wears a black hat at times to accomplish his own agenda (Attackers and vulnerabilities). The word “intent” calls to mind the Best Practices statement on plagiarism offered by the Council of Writing Program Administrators noting a student’s “attempt” to do the right thing. The problem is once again determining what constitutes “less than noble?” Should a student, for example, forego incorporating a portion of a digital text into their composition because despite diligent attempts to secure permissions they have failed to locate an owner? Due to the restructuring of copyright law, even what are known as “orphaned works” have a barrier of protection despite the absence of discernable owner. Of course it is part of a student’s “agenda” to use other texts as a means to enter a discourse community, to “create,” and to “say things differently” as they construct their own compositions. Indeed, these are our own pedagogical goals for writing and it seems futile to pretend that writing/hacking will not reside frequently in this gray area.

It may be a simple task to label someone a black hat hacker for installing spy-ware on an unprotected computer but when it comes to hacking as writing (which includes remixing a range of content that includes lines of code, digital images, video, prose, poetry and more) we are by the

very definition of text "grey" hat hackers. Relying on the work of other writers is what writers do. More eloquently, Porter describes the relationships between texts stating: "Not infrequently, and perhaps ever and always, texts refer to other texts and in fact rely on them for their meaning. All texts are interdependent: We understand a text only insofar as we understand its precursors" (1986, p. 34). With Web 2.0 we have come face to face with intertextuality and have found ourselves unprepared for the encounter. In what may be interpreted as a knee-jerk response, many institutes are now relying on plagiarism detection services despite excoriating reviews by scholars decrying their use. No tool will help us escape the fact that, "In instructional settings, plagiarism is a multifaceted and ethically complex problem" (Defining and Avoiding Plagiarism). Instead, a loose framework derived from the terms white ("good" hacking), black ("bad" hacking), and gray (ethically ambiguous hacking) can be an effective means to introduce students to Web 2.0 and the ethical concerns it brings without being prescriptive. As Howard remarked, "plagiarism-detecting software does not protect the learning experience; only pedagogy does" (2007, p. 11). The next section demonstrates how to use two Web 2.0 tools to situate student writing in ethically gray situations in order to facilitate a discussion on hacker ethics. Rethinking our pedagogy so we may engage with new forms of composition like Web 2.0 remixes as well as appropriate the open source community's notion of hacker ethics might, as Howard suggests, save us.

Firefox Extensions

The two Web 2.0 technologies discussed in this text, Web Developer and Greasemonkey, are "add-ons" or extensions for the free and open source web browser Firefox. An extension is a separate program created to add functionality to the browser. Since Firefox is open source, developers are able write extensions compatible with the browser because they are free to download, view, and study Firefox's source code. Firefox was designed and is still managed for extensibility or future growth by keeping its code open and creating a centralized location for developers and users to locate, download, comment on, and of course hack extensions. The

Firefox add-ons main page invites visitors to “Take a look around and make Firefox your own” (Firefox Add-ons). All of the extensions (there are thousands) may be downloaded and used “as is.” Extensions pages are set up with features similar to that of a blog where users can post comments about the extension, rate it, and even contact the primary developer responsible for the add-on to report a bug. More intrepid users can try their hands at re-writing the code for the extension to improve it themselves. However, most often the user’s creativity comes into play with their individual use of the new features afforded by the extension. As Wesch reminded us, we do not need to be professional programmers to participate in Web 2.0. For example, an extension like the Quick Locale Switcher enables a user to easily change between language settings for Firefox so the browser manages multilingual environments more efficiently including incorporating dictionary and spell-checking tools. How and in what contexts users make use of Quick Locale Switcher will vary greatly. This variance is apparent from the comments and reviews posted on its download page where one user extolled: “For all extension developers, installing this is an absolute requirement. This makes testing your localizations a breeze.” And for a seemingly altogether different use, another reviewer wrote: “My Japanese is much better as a result of using QLS. Thanks!” (Quick Locale Switcher)

Extension: Web Developer

Since the early days of the web, developers have taken advantage of a browser’s ability to “View Source” or display the underlying code that makes a web page possible. The resulting effect has been the promotion of a massive amount of knowledge sharing in regard to page construction and web development. With some time and patience, the answer to the question, “How did they do that?” can be discovered by essentially reverse engineering the page by viewing, copying, and editing a web page’s source code.

As the web grew and the programming possibilities expanded to incorporate the flexibility discussed by Wesch, web pages and their source code became exceptionally complex.

Wading through the source code attempting to deduce what piece of code was responsible for what functionality became an onerous task. Installing the Web Developer extension for Firefox adds a toolbar which enables a user to display visually the specific components comprising the construction of a web page. For example, among the Web Developer menu bar is a selection titled "Information." In this menu are options such as "Display Table Information," Display Anchors," "Display Id & Class Details," and "Display Div Order." The "Images" menu has selections enabling a user to, "Display Image Dimensions," Display Image File Size," and "Hide Images." These options and many others included in the Web Developer extension empower users to show, hide, and disable different stylistic as well as functional or interactive components of a site's design making it easier to begin remixing or rewriting the page. As one reviewer phrased it, "Its a fun form of web vandalism that doesn't affect everyone else. - And of course what it was meant for- Web Design" (Reviews for Web Developer).

It should be noted that the spirit of openness and the ability to "View Source" are not relished by every business online. For example, the below text is from a privacy and policy statement from the Dozier Internet Law firm which claims to be "The Lawyers for Internet Business."

> Dozier Internet Law, P.C. has a lot of intellectual property on our site. For instance, we are the creators of all of the text on this website, and own the "look and feel" of this website. We also own all of the code, including the HTML code, and all content. As you may know, you can view the HTML code with a standard browser. We do not permit you to view such code since we consider it to be our intellectual property protected by the copyright laws. You are therefore not authorized to do so. (User Agreement/Privacy Policy)

The language from the Dozier policy statement is stricter than most but as hackers/writers we will need to consider how statements like this affect our actions as we weigh ethical responsibilities before rewriting a page. I believe the contrast created by juxtaposing protectionist policy statements and Web 2.0's unregulated approach to accessing data is an effective teaching tool in

the writing classroom because it brings students face-to-face with conflicting value systems. This is the point made by Adler-Kassner, Anson, and Howard when they wrote: “Reclaiming education entails teaching students to recognize and adapt to wide variations in the values that determine how a text is created, used, and represented in specific social, academic, and occupational contexts – values often connected to cycles of credit and credibility that obtain in the academy and the larger culture” (2008 p. 236).

Extension: Greasemonkey

The first step to remix then is for the hacker to decide what he or she would like to alter on an existing web page and then use the Web Developer extension to understand how the page is constructed. To facilitate rewriting a web page, hackers can download and use the Greasemonkey extension for Firefox. Greasemonkey enables users to compose and run “action scripts” which are designed to load in conjunction with the source code for the page being altered. The scripts are written in the common JavaScript language and are “client-side” scripts. This means that the scripts run on the user’s own computer only and do not require a hack into the server hosting the altered page. The alterations or hacks take place as the page is loaded in the user’s browser and render without the consent of the site owners. Greasemonkey enables users to take on the roles of co-author and even compose an “oppositional discourse” mentioned by Selber. In the coming examples I will demonstrate how to write two simple Greasemonkey scripts. However, it should be noted that there are hundreds of scripts already in existence and posted for downloading on sites such as userscripts.org. These scripts do anything from stripping advertisements out of specific web sites like Yahoo! to using Amazon’s powerful and extensive book database to search and buy books from Amazon’s competition.

Sample Hacks

The range of perspectives offered for how we understand "hacker ethics" including those from original open source activists like Raymond, definitions from Red Hat's Linux Security Guide, and the more rigid guidelines found in the preparation manuals for becoming a "certified" ethical hacker make for a complex composite term. In short, these sources tell us that hacker ethics has everything to do with access, conduct, and motivation.

Raymond's contributions suggest that if a programmer wishes to be considered part of the hacker community then he or she must endeavor to share their work and keep access to their code open. Access means more programmers can begin building and improving on another hacker's progress without duplicating their efforts. Within the hacker community there is "the powerful sense that it is not merely inefficient but downright stupid, almost criminal, for people to have to solve the same problem twice" (Weber 138). Problems, even criminal ones, can come about when that access is assumed but not granted. The certified ethical hacker manuals instruct that nothing be hacked without permission. Conversely, Raymond contends that "system cracking for fun and exploration" falls into the realm of "ethically OK" as long as no harm is done. Meanwhile, the Red Hat security guide informs that any hacking without those permissions automatically classifies our conduct as "grey hat" hacking. So why hack? Raymond finds that motivation for our conduct is spurred by pure "intellectual curiosity" or the need for "circumventing limitations." Hacking, according to Himanen, is most often "an activity that is intrinsically interesting, inspiring, and joyous" (6). Our perceived sense of satisfaction may come from contributing to solving a communal problem or it may "scratch a personal itch." Of course, our desire to learn how something works or even improve on an existing program or web page does not automatically provide a legal justification for our conduct. Either way, designed for "hackability" and "remixability," Web 2.0 places us squarely in the middle of a very gray space. Below are two modest, sample hacks that employ the Web Developer and Greasemonkey extensions for Firefox and may be used to bring students into discussions regarding the ethics of remixing and rewriting an interface and its functionality.

Sample 1: Web Based Email Log-In

The first example addresses my own frustrations with logging-in to my university's web-based email system. If I clear my browser's cache the system does not remember my email user name and the cursor is never positioned inside of the password field waiting for me to type. The company Novell provides our email so I visited their corporate site to read their legal policy statements:

> The design or layout of the Novell.com website or any other Novell owned, operated, licensed or controlled site is the property of Novell, Inc. Elements of Novell websites are protected by copyright, trade dress and other laws and may not be copied or imitated in whole or in part. (Novell)

Given this policy statement, I like to ask my students in classes like technical communication or multimedia writing to consider arguments on whether or not it would be ethical to use Web 2.0 technologies like Web Developer and Greasemonkey to hack the functionality of the log-in page? The "white hat" position would suggest that hacking copyrighted material of any kind is verboten while the "black hat" position would not hesitate to do so. As I have suggested with this new form of composition, the practice is inevitably gray. I do, as the Red Hat definition of a "grey hat" states, have my own "agenda" for the page. While I am not hacking into an email server, rewriting the functionality of Novell's interface is certainly not what their policy statement invites. The question of ethics becomes more complex when we consider whether or not I should distribute the script to other frustrated colleagues or even post the script to a site such as userscripts.org. Inevitably someone suggests that copyright law's provision for Fair Use should protect our actions were we to be prosecuted and that should put our ethical concerns to rest. However, interpreting what constitutes Fair Use is just as "grey" as our ethical debate. Even if I was to argue for the educational use of the script it is impossible to predict whether or not this hack would be interpreted as having a negative impact on Novell's place in the market. Who, after all, could afford to go to court to defend a Fair Use claim?

Research by scholars like Adler-Kassner, Anson, Howard, Porter, Selber, and Giesler, et al. writing on intertextuality provides the encouragement to move ahead with writing the script – not Fair Use. Likewise, recent publications such as George Pullman's (2005) "From Wordsmith to Object-Oriented Composer" and Bill Hart-Davidson's (2005) "Shaping Texts That Transform: Toward a Rhetoric of Objects, Relationships, and Views," which have included detailed code and addressed object-oriented programming and its relationship to writing, have encouraged me to include code in this text. Pullman's and Hart-Davidson's publications contain explications for lines of XML as well as HTML forms and the use of more challenging Perl and CGI scripts. As Pullman remarked, "Believe me, if I can do this kind of thing, anybody can" (p. 57). I am including the code for my Greasemonkey scripts below because I feel the same way about using these new Web 2.0 technologies. Although mastering Web Developer, Greasemonkey, or computer languages like JavaScript are not required to introduce students to white, black, and gray hat ethics, we should not shy away from engaging these new technologies. Below in Figure 1 is a screen shot of the login page for my university's web-based email.

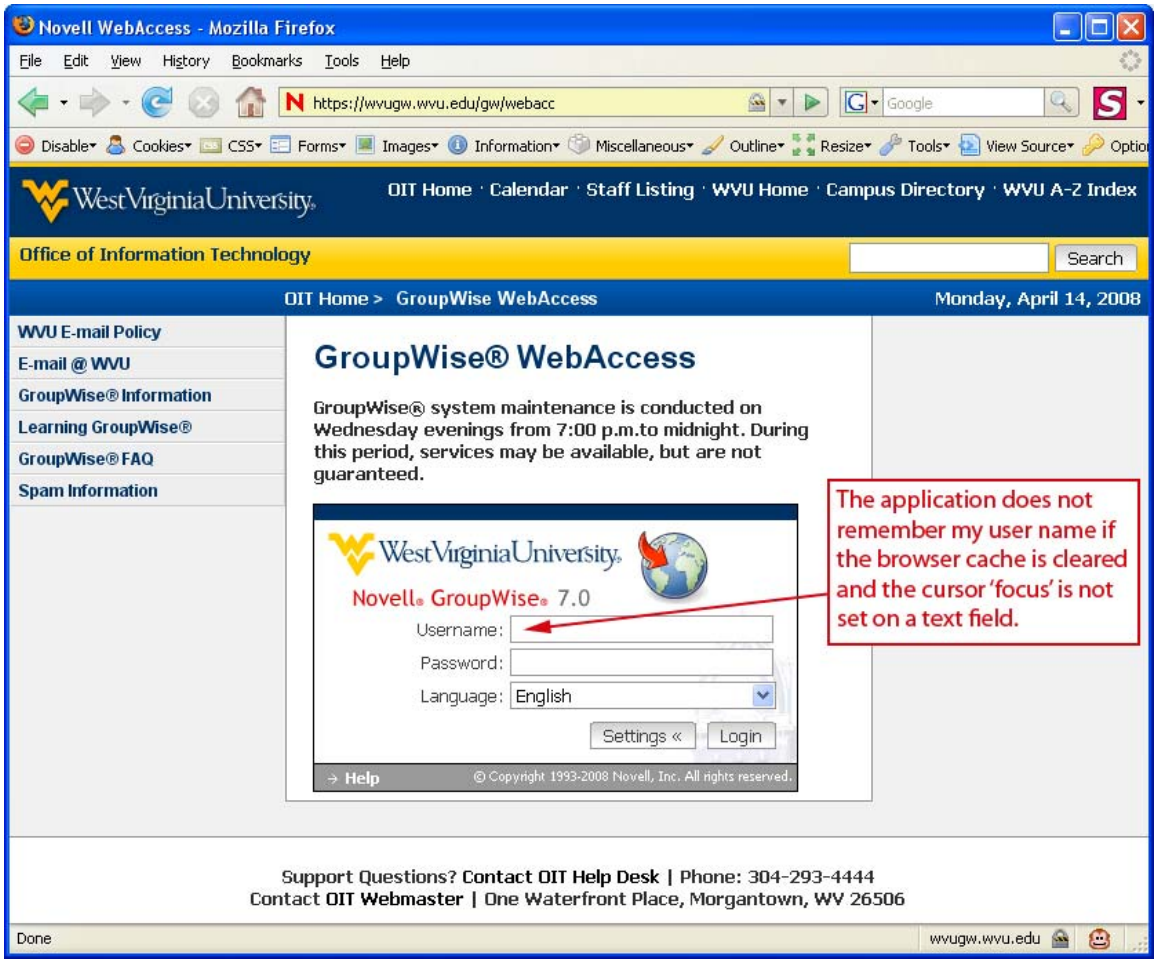

Figure 1: West Virginia University's web-based email login page.

Using Web Developer's tool bar, a user would click "Forms" and select the first menu item "Display Form Details" as shown in Figure 2.

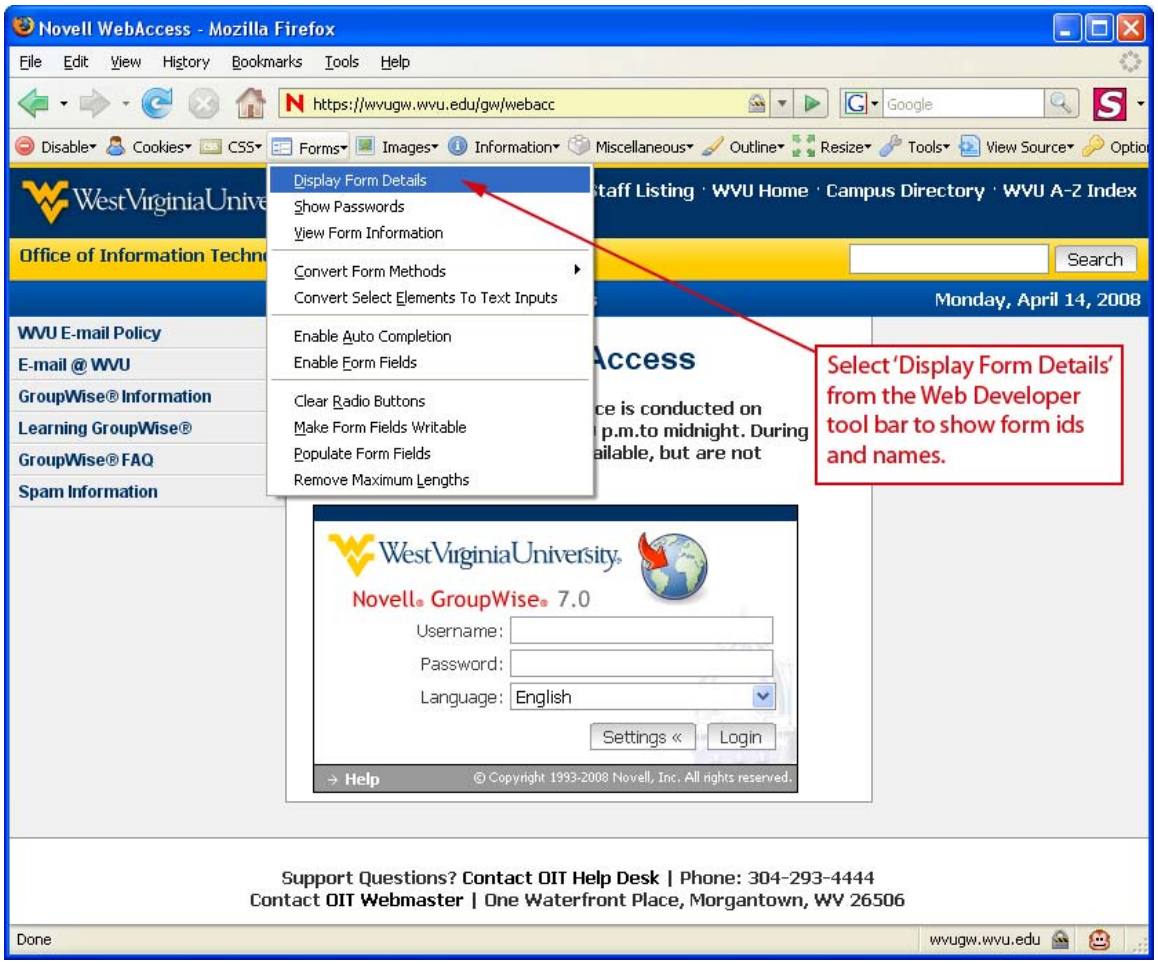

Figure 2: The 'Forms' button is clicked from the Web Developer toolbar and the 'Display Form Details' option is selected.

The results from selecting "Display Form Details" are shown below in Figure 3.

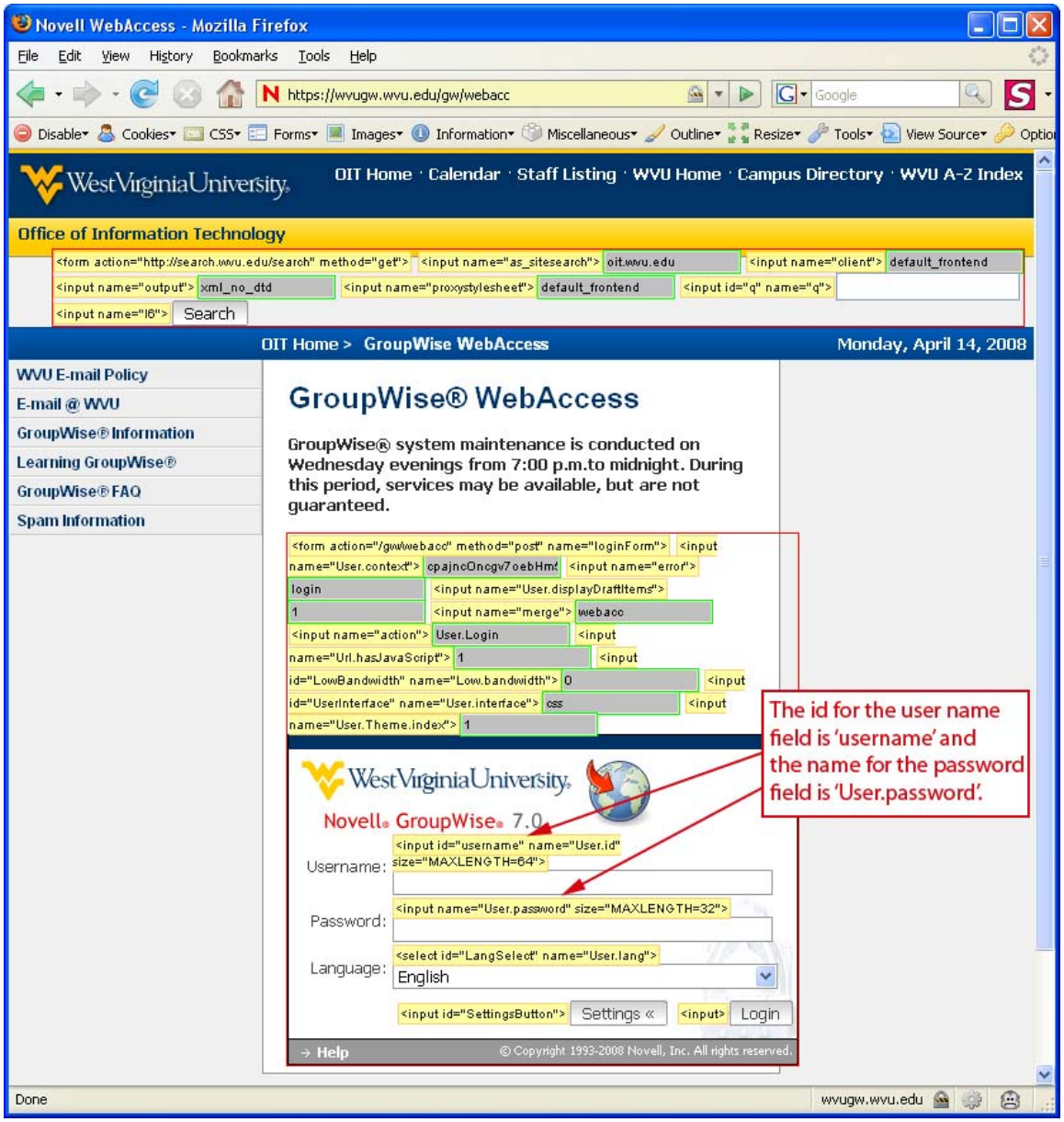

Figure 3: Web Developer displays information on the Username and Password text fields.

Web Developer will display information on any of the HTML forms on the page including the search function at the top of the screen. To write the hack, we need information on the text fields for the user name and password. Web Developer shows us that the “id” for the user name is simply “username” but that the password field does not have an id. We will use the field’s

“name” to manipulate the page and that value is “User.password”. This is enough information to begin writing our Greasemonkey script.

Scripts can be composed with any word processing application or HTML editor. The only stipulation is that the file is saved with the “user.js” suffix. The word “user” makes the file recognizable as a Greasemonkey script and the “js” stands for JavaScript. In this example, my filename is “GW-autocomplete.user.js”. In order to install the script, Greasemonkey must first be downloaded and installed. Afterwards, a user simply clicks on the "user.js" file to load the script. This script will automatically load when I visit the login page for my university email. The code is centered on one basic JavaScript function that contains two variables. The first variable that I named “login” is used to automatically assign a value to the user name field in the form. I can do this because I used Web Developer to discover the id of the field which is “username.” With the script, I declare the value of “username” as my own user name or “BDBallentine.” The second variable I named “setit”. Again, we know from Web Developer that the “name” for the password field is “User.password”. All I wish to do is make or “set” the cursor flashing inside of this field so I can type my password and quickly login. This means setting what is known as the “focus” for the cursor on a particular field. Without going into all of the intricacies, the complete script appears as below:

```
// ==UserScript==
// @name           Auto-complete
// @namespace      https://wvugw.wvu.edu/gw/webacc
// @description    Auto-completes the username login field on Novell GroupWise web login
// @include        https://wvugw.wvu.edu/gw/webacc
// ==/UserScript==

(function()
{
        //look for the username login field
        var login = document.getElementById('username');
        if(login != null)
        {
                //write your username to the field
                login.value = 'BDBallentine';
        }
//set the cursor focus in the password field
```

```
var setit = document.getElementsByName('User.password');
        if(setit != null)
        {
                setit[0].focus();
                }
})();
```

After the script is installed, each subsequent visit to the web-based email login page yields the desired result as shown below in Figure 4.

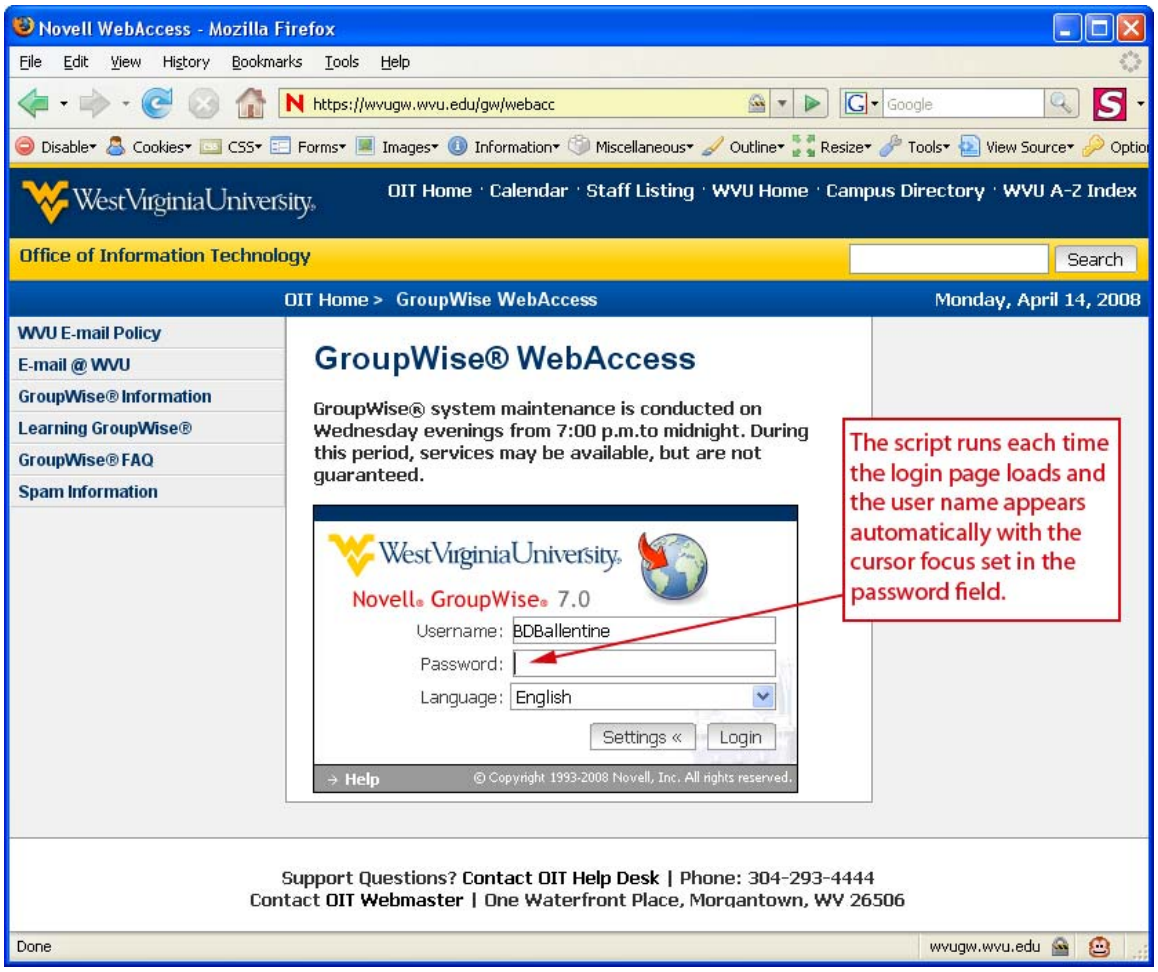

Figure 4: The final result of the Greasemonkey script.

Student response to a hack such as this is almost overwhelmingly positive and enthusiastic. Selber's concept of critical literacy or our opportunity to "engage in micropolitical acts of modification that adapt technologies to users" becomes a modest, but demonstrable reality in the

classroom (p. 105). The discussion of ethics was more robust not just because we appropriated the framework of white, black, and gray hat hackers but because students had a hand in or at least were witness to the activity of re-writing the interface.

Sample 2: CNN Special Reports

For the second example, the idea was to increase the focus on ethics by debating not just our own actions but the actions of the web site I was considering we hack. In the Health section of CNN's news web site the company offers a special report titled "Matters of the Heart." The page invites readers to learn "about the latest advances in medicine, and get tips and strategies to stay healthy, eat well and exercise right" (Matters of the Heart). Featured prominently at the top of the page is a rotating advertisement that often features the cholesterol drug Crestor as seen in Figure 5. The advertisement fills nearly a third of the top heading for the special report. Indeed it appears to some to be part of the report. In very small print CNN includes the word "advertisement." Of course CNN like any news source needs to generate revenue and this often accomplished by way of advertising. I remind the students that just hosting the CNN web site is costly not to mention paying the content writers and web developers. However, when I query my students about the ethics of the placement and even the inclusion of the advertisement most take issue with its tight integration with the page and some even find fault with its presence in what is labeled a "Special Report."

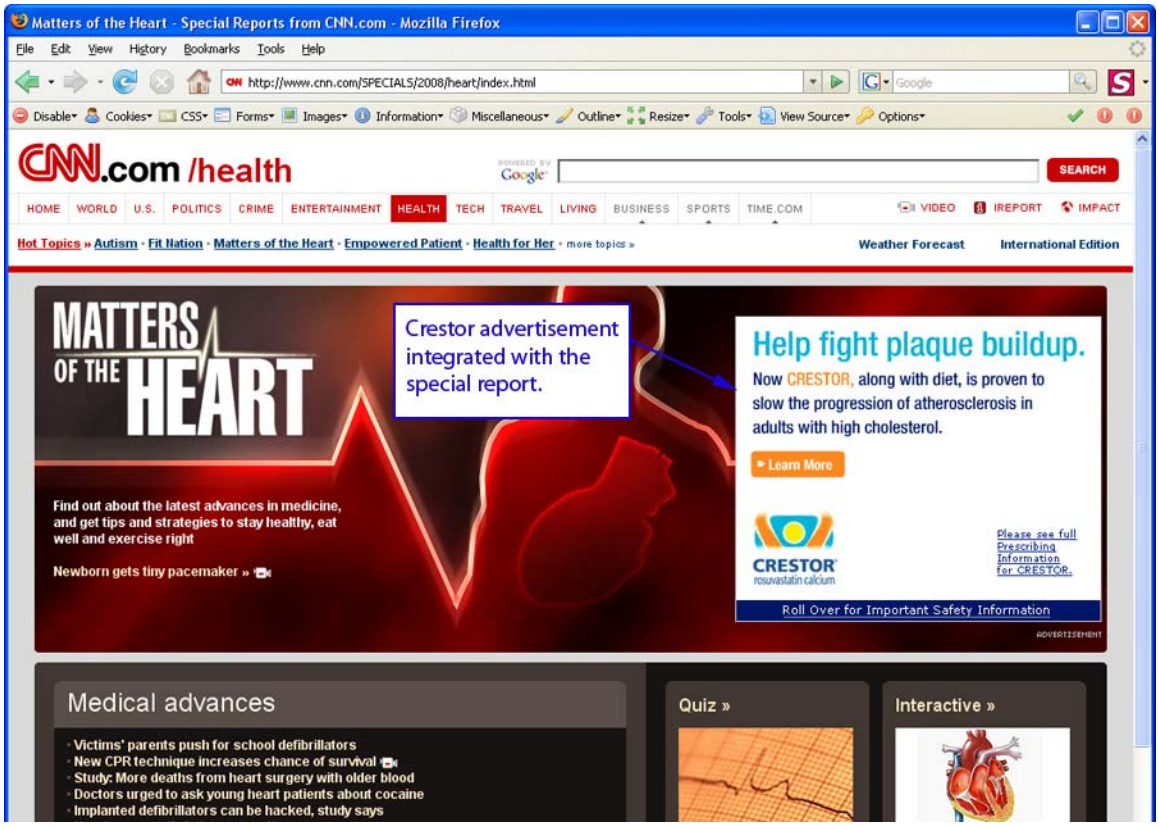

Figure 5: CNN's Special Report on "Matters of the Heart" with Crestor advertisement showing.

Next, I give the class the option of removing the Crestor advertisement with a Greasemonkey script but the students can not agree that the removal is either a purely white or black form of ethical hacking. Our debate over the “grey-ness” of ethically hacking out the advertisement leads the class to examine the advertising section of CNN. The students deduce that Crestor has paid for one of the larger and no doubt more expensive advertising placements CNN has to offer (Advertise: Ad Specs & Guidelines). To entice advertisers, CNN boasts 1.6 billion monthly page views among which they have identified 28.4 million unique users (Advertise: Audience Profile). There is little question that neither CNN nor Crestor would be receptive to the Web 2.0 spirit of openness that invites co-authorship of the special report especially when that co-authorship involves removing expensive advertisements. My students, however, decided differently and after

their dialog regarding the CNN site they determined that the advertising strategy warranted the "technological reconstitution" of the page (Selber 105).

Once again using Web Developer's tool bar, a user can click on "Information" and select "Display Id & Class Details."

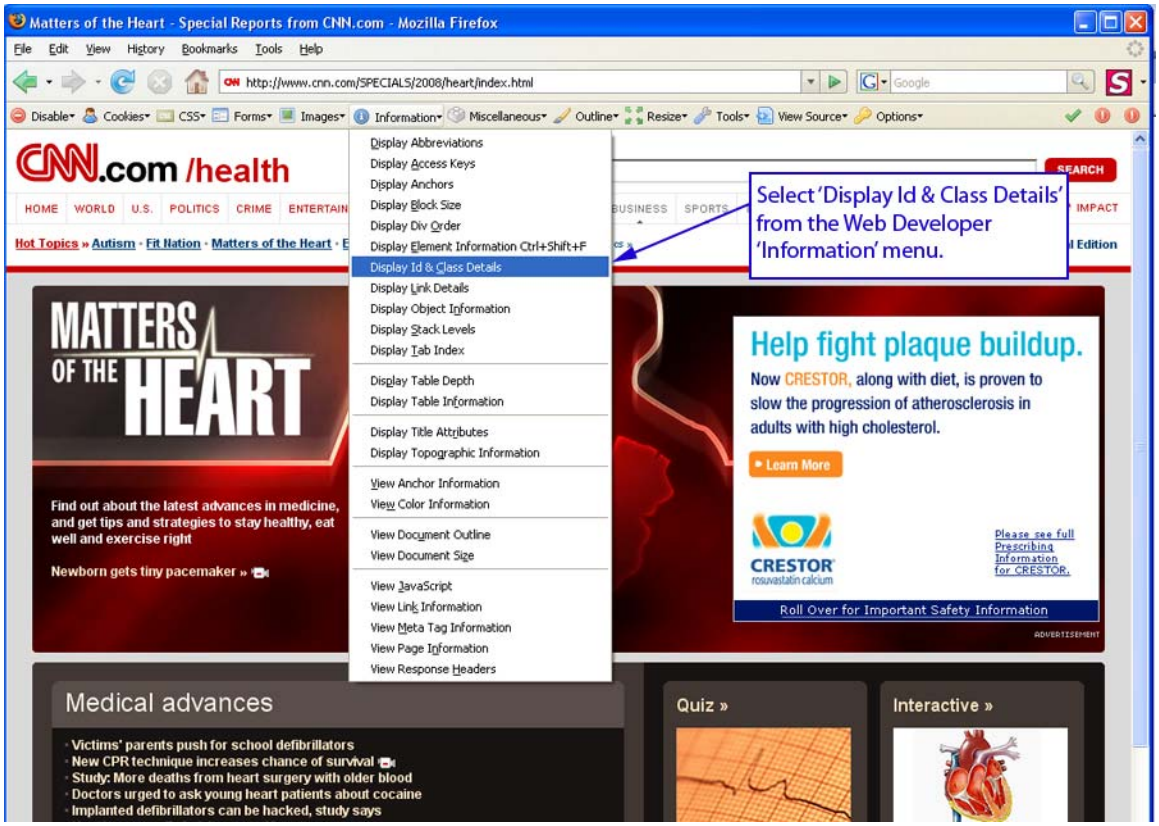

Figure 6: The 'Information' button is clicked from the Web Developer toolbar and the 'Display Id & Class Details' option is selected.

CNN uses a complex structure of <table> and tags to format their pages and Web Developer makes the process of identifying the id associated with the ad easy. The Crestor ad is positioned inside of a tag with the id number "ad-571686" as shown below in Figure 7.

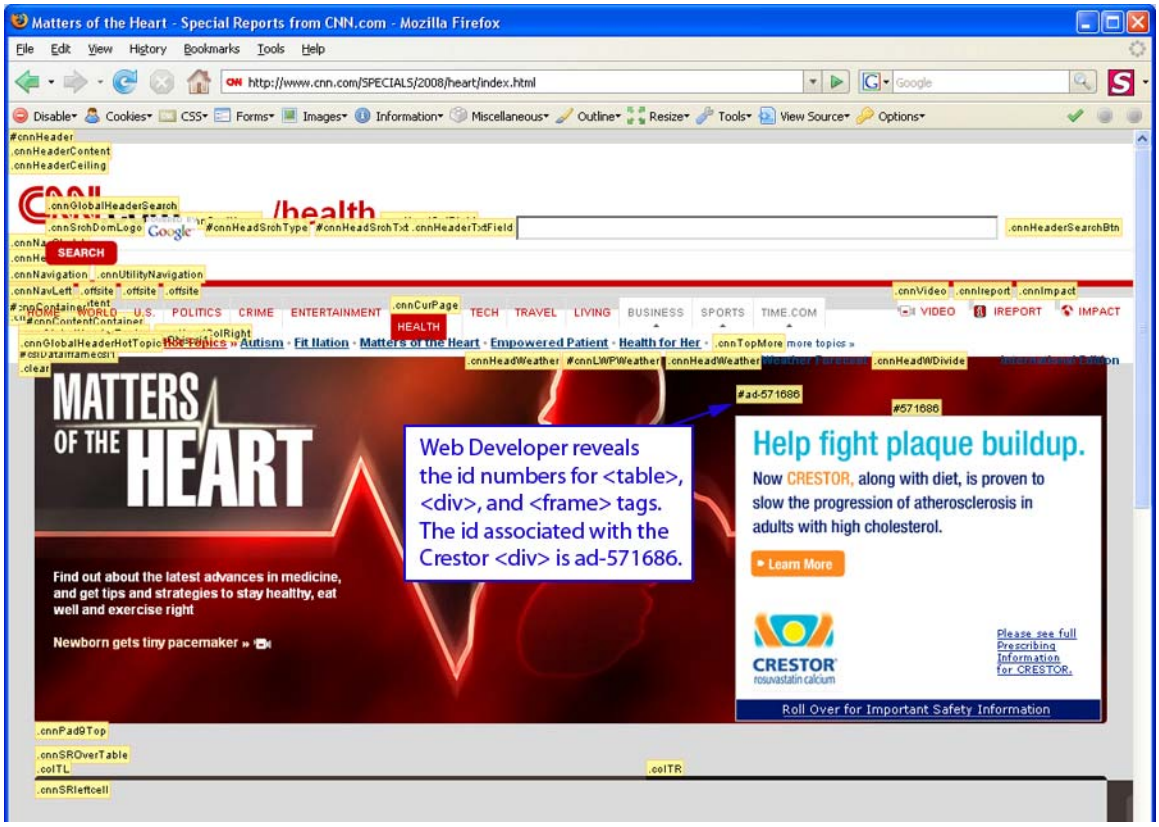

Figure 7: Web Developer displays information on the Crestor ad Div id.

With a few simple lines of code we can compose a Greasemonkey script hides anything displayed inside of the ad-571686. This script named CNN-HeartAd.user.js uses a JavaScript function to dynamically hide the contents of a specific on CNN's special report. I created a variable titled "findFrameID" which was aligned with the tag's id. The last step was to set the "display" of that variable to "none" thus hiding the Crestor ad.

```
// ==UserScript==
// @name           CNN Ad Remover
// @namespace      http://www.cnn.com/SPECIALS/2008/heart/index.html
// @description    Strips ad from top right corner
// @include        http://www.cnn.com/SPECIALS/2008/heart/index.html
// ==/UserScript==

(function()
{
// assign the Frame or Div ID
        var findFrameID = document.getElementById("ad-571686");
        if(findFrameID != null)
```

```
        {
                //set style of iFrame to hidden
                findFrameID.style.display = "none";
        }
})();
```

Readers who have downloaded Greasemonkey can try out this script by installing it and then visiting the CNN special report page on Matters of the Heart. Note that if CNN has changed or renamed any of the relevant information the script will cease to function and need to revised. The final product is below in Figure 8.

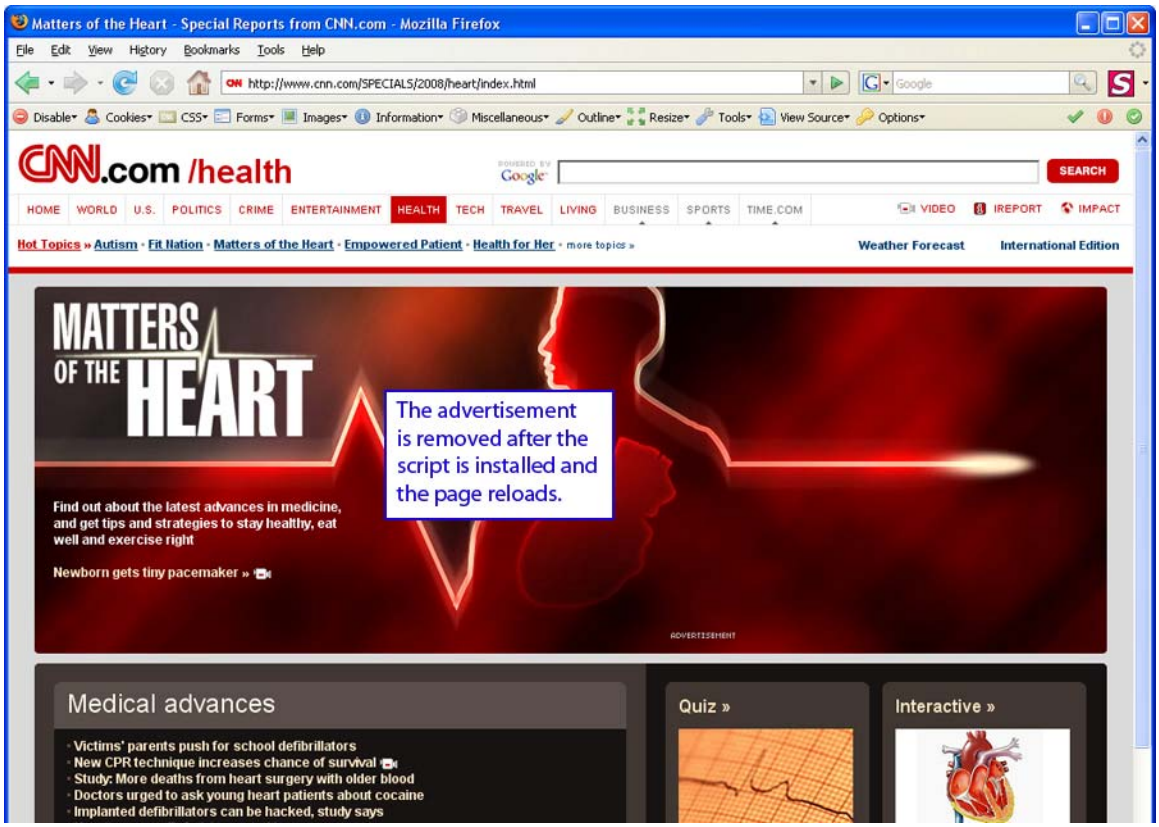

Figure 8: The final result of the Greasemonkey script removing the Crestor ad.

According to Raymond, “Every good work of software starts by scratching a developer's personal itch” (Cathedral and the Bazaar). Web 2.0 empowers users to do just that. These small sample hacks are an effective means to bring students into discussions regarding the ethics of remixing and rewriting an interface.

Closing

> Central to the idea of rhetorical theory is the idea that audience determines the appropriateness and success of communication. The expectations, task needs, immediate situation, values, interests, and presumptions of readers affect how they understand and respond to a text or message (Giesler et al., p. 271-2).

Web 2.0 has broadened significantly the available means of recourse for an audience that finds the "appropriateness" or "success" of an IText lacking. The breadth of options also presents instructors with a unique opportunity to bring ethics into the classroom. In Edward McQueeney's "Making Ethics Come Alive," he remarks on students in his communication course who "often perceive ethics to be spongy, soft, and largely irrelevant to the skills necessary to do business" (p. 158). The problem McQueeney finds is the "air of unreality that often surrounds classroom ethics discussion" (p. 158). I believe that McQueeney is not alone in his experiences with attempting to bring ethics into a writing classroom of any kind. Case studies and even examining existing codes of ethics are useful tactics for engaging students but with Web 2.0 we can confront students with ethically "grey" scenarios regarding remixing the web. This pedagogical approach to ethics deliberately prompts ethical debate without prescribing right or wrong. A neutral stance is an important part of my own strategy for moving class discussion from two specific case studies to some more general ethical concerns of writers. As Donna Kienzler (2001) remarked, "A critical thinking perspective also demands an ethical instructor. One of the first demands on that instructor is a safe learning community" (p. 325). With the two examples above, I strive to create an open environment where the "right" answer is up for debate and it is in the process of forming arguments and thinking critically about options that students learn about ethics. The approach also brings instructors face-to-face with the complicated relationship between writing, intertextuality, and the debate over what constitutes plagiarism described by Howard and others.

In addition, to expand our exploration of ethics, I again use McQueeny's article where he states, "Ethical behavior costs something, and its consequences, to the individual as well as to the organization, can be very serious" (p. 160). I then posit his remark in the form of a question for the class. What are the costs of white hat hacking where permission for the use of every source is required? Conversely, what are the potential costs of black hat behavior where no effort is made to secure any permission? Our two case study hacks with Web Developer and Greasemonkey are good points of departure for widening our discussion of ethical behavior into other writing situations. For example, what are the costs of exaggerating experience in a résumé? Would you classify that behavior as white, gray, or black hat hacking? Likewise, what are the potential costs in including a "drop dead" implementation date within a business proposal that can not be guaranteed with 100% certainty? At what point, if any, is including such a date acceptably "grey"? In fact, I have found the hacker color spectrum to be applicable and useful for discussing a variety of writing situations and the Web 2.0 technologies Web Developer and Greasemonkey to be excellent tools for introducing students to the hacker community.

Most likely the concerns writing instructors will have with attempting to integrate Web 2.0 technologies and hacker ethics into a writing course will be in regard to the learning curves associated with tools like Web Developer and Greasemonkey. These tools like many others offered in Web 2.0 communities allow users to wield them with varying degrees of knowledge and skill. There are, for example, many Greasemonkey scripts ready to download from userscripts.org that would function equally well as prompts for classroom conversations on ethical hacking. Since userscripts.org facilitates the dissemination of scripts that do everything from remove expensive advertisements to remixing content that was never intended to be combined, use of the site and even its existence is an interesting point of entry for discussing ethics.

I would also suggest that most of the apprehension over using these new technologies will not be found with the students. After listing numerous emerging genres resulting from

compositions in digital spaces, Yancey remarks that students are already motivated to embrace the challenges of working with new media.

> Note that no one is making anyone do any of this writing. Don't you wish that the energy and motivation that students bring to some of these other genres they would bring to our assignments? How is it that what we teach and what we test can be so different from what our students know as writing? (p. 298)

This is not to say that in order to curry favor with students we should abandon all existing writing pedagogy and move exclusively to online environments. I believe we should continue to teach our writing courses emphasizing the genres and principles of conduct our fields recognize as valuable. In writing courses that range from introductory composition to multimedia writing, ethics, in some form, usually makes the short list. What I am suggesting here is that hacker ethics has a unique relationship to writing in digital environments that we can leverage to help us navigate our foray into writing with Web 2.0.

References

About. *Creative Commons*. Retrieved April 30, 2008, from http://creativecommons.org/about

Adler-Kassner, Linda, Anson, Chris, and Howard, Rebacca Moore. (2008). Framing plagiarism. In Caroline Eisner and Martha Vicinus (Eds.), *Originality, Imitation, and Plagiarism: Teaching Writing in the Digital Age*. (pp. 231 – 246). Ann Arbor: U. of Michigan Press.

Advertise: Ad specs & guidelines. *CNN*. Retrieved April 14, 2008 from http://www.cnn.com/services/advertise/specs/specs_overview.html

Advertise: Audience profile. *CNN*. Retrieved April 14, 2008 from http://www.cnn.com/services/advertise/audience_profile.html

Attackers and vulnerabilities. (2002). *Red Hat Linux 9: Red Hat Linux security guide*. Retrieved Nov. 4, 2007, from http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/ch-risk.html

Ballentine, Brian. (2007). Greasemonkey and a challenge to the notions of authorship. In Kirk St. Amant & Brian Still (Eds.), *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*. (pp. 12-22). Hershey, PA: Idea Group.

Defining and avoiding plagiarism: The WPA statement on best practices. (2003). *Council of Writing Program Administrators*. Retrieved Jan. 12, 2009, from http://wpacouncil.org/positions/WPAplagiarism.pdf

Davydov, Dmitri. (2007). Little known 'boring' websites that make incredible money with AdSense. *Nichegeek.com*. Retrieved Feb. 15, 2008, from http://nichegeek.com/little_known_boring_websites_that_make_incredible_money_with_adsense

Firefox add-ons. *Mozilla Firefox*. Retrieved March 1, 2008, from https://addons.mozilla.org/en-US/firefox/

Geisler, Cheryl, Bazerman, Charles, Doheny-Farina, Stephen, Gurak, Laura, Haas, Christina, Johnson-Eilola, Johndan, Kaufer, David S., Lunsford, Andrea, Miller, Carolyn R., Winsor, Dorothy, & Yates, JoAnne. (2001). IText: Future directions for research on the relationship between information technology and writing. *Journal of Business and Technical Communication*, 15(3), 269-308.

Goetz, Thomas. (2003). Open source everywhere. *Wired*, 11(11), 160-167, 208, 210-11.

Gregg, Michael. (2006). *Certified Ethical Hacker*. Indianapolis, IN: Que Publishing.

Gurak, Laura J. (2001). *Cyberliteracy: Navigating the Internet with Awareness.* New Haven, CT: Yale U P.

Harris, Shon, Harper, A., Eagle, C., Ness, J. & Lester, M. (2005). *Gray Hat Hacking: The Ethical Hacker's Handbook.* New York: McGraw-Hill.

Hart-Davidson, Bill. (2005). Shaping texts that transform: Toward a rhetoric of objects, relationships, and views. In Carol Lipson and Michael Day (Eds.), *Technical Communication and the World Wide Web*. (pp. 27-42). Mahwah, NJ: Lawrence Erlbaum.

Himanen, Pekka. (2001). *The Hacker Ethic and the Spirit of the Information Age*. New York: Random House.

Howard, Rebecca Moore. (2007). Understanding 'internet plagiarism.' *Computers and Composition*, 24, 3-15.

Howard, Rebecca Moore. (1999). *Standing in the Shadow of Giants: Plagiarists, Authors, Collaborators*. Stamford, CT: Ablex.

Kienzler, Donna. (2001). Ethics, critical thinking, and professional communication pedagogy. *Technical Communication Quarterly*, 10(3), 319-339.

Legal & Export. *Novell*. Retrieved April 2, 2008, from http://www.novell.com/company/legal/

Lessig, Lawrence. (2007). How creativity is being strangled by the law. Presentation at the Technology Entertainment and Design Conference (TED). Monterey, CA. Retrieved March 18, 2008, from http://www.ted.com/talks/view/id/187

Lessig, Lawrence. (2005). Remix culture. Featured Session at the Conference on College Composition and Communication. San Francisco, CA. Retrieved December 7, 2007, from https://netfiles.uiuc.edu/jlsolber/www/lessig

Lessig, Lawrence. (2004). *Free Culture: The Nature and Future of Creativity*. New York: Penguin Books.

Manjoo, Farhad (2008, March 21). The Wall Street Journal's web site is already (secretly) free. *Salon.com*. Retrieved Aug. 2, 2008 from http://machinist.salon.com/blog/2008/03/21/wsj/index.html

Matters of the Heart. *Special Reports from CNN*. Retrieved April 14, 2008 from http://www.cnn.com/SPECIALS/2008/heart/index.html

McHugh, Josh. (2008). The data wars. *Wired*, 16(1), 136 – 139, 159.

McQueeney, Edward. (2006). Making ethics come alive. *Business Communication Quarterly*, 69(2), 158-171.

NCTE – CCCC – 2008 Resolutions (2008). Retrieved Aug. 30, 2008 from http://www.ncte.org/cccc/resolutions/2008

Oliu, Walter E., Brusaw, Charles T., & Alred, Gerald J. (2007). *Writing That Works: Communicating Effectively on the Job*. New York: Bedford/St. Martins.

O'Reilly, Tim (2005, September 30). What is Web 2.0? Design patterns and business models for the next generation of software." *O'Reilly.com*. Retrieved October 1, 2006, from http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html

Porter, James. (1998). *Rhetorical Ethics and Internetworked Writing.* Greenwich, CT: Ablex.

Porter, James. (1986). Intertextuality and the discourse community. *Rhetoric Review*, 5(1), 34-47.

Pullman, George. (2005). From wordsmith to object-oriented composer. In Carol Lipson and Michael Day (Eds.), *Technical Communication and the World Wide Web*. (pp. 43-59). Mahwah, NJ: Lawrence Erlbaum.

Raymond, Eric (2000). *The Cathedral and the Bazaar*. Retrieved May 1, 2008, from http://catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html

Raymond, Eric. *The Jargon File*. Retrieved May 9, 2008, from http://catb.org/~esr/jargon/html/frames.html

Reviews for Quick Locale Switcher: Firefox Add-ons. *Mozilla Firefox*. Retrieved March 1, 2008, from https://addons.mozilla.org/en-US/firefox/reviews/display/1333

Reviews for Web Developer. Mozilla Firefox. Retrieved March 1, 2008, from https://addons.mozilla.org/en-US/firefox/reviews/display/60

Salvo, Michael. (2005). Teaching information architecture: Technical communication in a postmodern context." In Carol Lipson and Michael Day (Eds.), *Technical Communication and the World Wide Web*. (pp. 61-79). Mahwah, NJ: Lawrence Erlbaum.

Selber, Stuart. (2004). *Multiliteracies for a Digital Age*. Carbondale, IL: Southern Illinois Univ. Press.

Selfe, Cynthia. (1999). *Technology and Literacy in the Twenty-First Century: The Importance of Paying Attention*. Carbondale, IL: Southern Illinois Univ. Press.

Stallman, Richard. (1999). The GNU operating system and the free software movement. In Chris DiBona, Sam Ockman, & Mark Stone (Eds.), *Open Sources: Voices from the Open Source Revolution*. (pp. 53-70). Sebastopol, CA: O'Reilly.

Thaiss, Christopher. (2001). Theory in WAC: Where have we been, where are we going?" In Susan McLeod, Eric Miraglia, Margot Soven, & Christopher Thaiss (Eds.), *WAC for the New Millenium: Strategies for Continuing Writing-Across-the-Curriculum Programs*. (pp. 299-325). Urbana, IL: NCTE.

Tidwell, Jenifer. (2006). *Designing Interfaces*. Sebastopol, CA: O'Reilly.

User Agreement/Privacy Policy. *Dozier Internet Law, P.C.* Retrieved March 1, 2008, from http://dozierinternetlaw.cybertriallawyer.com

Weber, Steven. (2004). *The Success of Open Source*. Cambridge, MA: Harvard UP.

Wesch, Michael. (2007). The Machine Is Us/ing Us. Retrieved Jan. 15, 2008, from http://youtube.com/watch?v=NLlGopyXT_g

Wheeler, Steve. (2001). Information and communication technologies and the changing role of the teacher. *Journal of Education Media*, 26(1), 7-17.

Yancey, Kathleen Blake. (2004). Made not only in words: Composition in a new key. *College Composition and Communication*, 56(2), 297-328.